Semester 2 Examinations 2014/2015

Exam Code(s)	4BCT
Exam(s)	B.Sc. in Information Technology
Module Code(s)	CT420
Module(s)	Real-Time Systems
Paper No.	1
Repeat Paper	N
External Examiner(s)	Dr. John Power
Internal Examiner(s)	Prof. Gerard Lyons Dr. Michael Madden *Dr. Hugh Melvin *Dr. Michael Schukat

Instructions:

Answer Q1 and 1 other Q from section A.
Answer 2 questions from section B.
All questions carry equal marks.

Duration	2 hours
No. of Pages	8
Discipline	Information Technology

Requirements:

MCQ
Handout
Statistical/ Log Tables
Cambridge Tables
Graph Paper
Log Graph Paper
Other Materials

Release to Library: Yes ☒ No □

Section A

Q1

(i) Explain using examples why many hard RTS use a cyclic executive approach to scheduling whereas many soft RTS utilise a real-time operating system.

[10]

(ii) Using Antilock Braking Systems (ABS) as an example, distinguish between the *sample rate* and *response time* of a RTS.

[10]

(iii) Quality of Experience (QoE) is a related though much broader concept than Quality of Service (QoS). Distinguish between the two terms using MMOG (Massively Multiplayer Online Gaming) as an example.

[10]

(iv) Distinguish between narrowband (NB), wideband (WB), and super-wideband (SWB) voice codec's, outlining the advantages/disadvantages of each.

[10]

Q2

(i) Distinguish briefly between **time and timing synchronisation**, using Voice-over-IP (VoIP) as an example.

[10]

(ii) Using each of the following examples, explain firstly - whether time, timing, or both types of synchronisation are important and secondly - outline the level of synchronisation required:

- Electrical Power Phasor Measurement Units (PMU)
- Fault diagnosis in a Distributed Control System
- MMOG (Massively Multiplayer Online Gaming) Server servicing players from across the globe

[15]

(iii) The data below shows the output of the **ntpq** utility from 2 NTP clients, A and B.
Based on the output, which client is most likely to deliver better Synchronisation? Your answer should comment on redundancy, offsets, delay, stratum level and reachability.

ClientA	RefID	st	t	When	Poll	Reach	Delay	Offset	Jitter
+server1	serverx	2	u	51	64	156	391.281	6.24	7.79
-server 2	servery	2	u	49	64	372	353.217	10.435	1.663
*server3	serverz	2	u	50	64	356	85.688	2.465	2.666
+server4	.PPS.	1	u	49	64	357	66.369	1.858	2.105
Client B	**RefID**	**st**	**t**	**When**	**Poll**	**Reach**	**Delay**	**Offset**	**Jitter**
+server5	serverm	2	u	45	64	377	36.345	2.24	3.458
-server 6	servern	2	u	42	64	377	313.217	3.435	1.11
*server7	.PPS.	1	u	13	64	377	10.688	1.485	1.455
+server8	.PPS.	1	u	22	64	377	28.456	1.818	1.345
+server9	.GPS.	1	u	18	64	356	15.345	-2.654	2.34

[15]

Q3

(i) As a senior network designer for an IP communications provider, you are asked to outline a complete end-to-end infrastructural design for a potential Irish client company that wishes to implement an all-IP based conferencing (voice/video) system. The company will have two large facilities, in Dublin and Galway, each with their own LAN. Your design should address the following issues:

- How to choose VoIP phones to maximise voice quality
- How QoS will be achieved across company LANs
- How QoS will be achieved across WAN. You can assume that your company uses DiffServ in its IP network.

[25]

(ii) MPEG-DASH and WebRTC are two recent developments in multimedia. Explain briefly what they are, how they work, and the extent to which time and or timing are important.

[15]

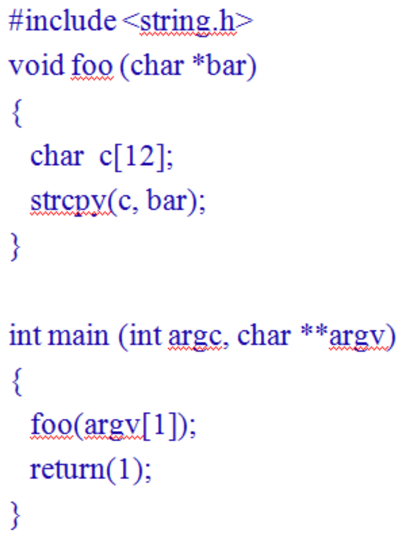
Section B

Q4

(i) Briefly distinguish between (a) deeply embedded systems (ES), (b) shallow ES, (c) medium ES, (d) commodity ES and (e) critical ES. Provide examples for any 4 of these categories that can be found in a car.

[6]

(ii) Using the source code snippet below show in detail how a stack overflow can be caused and how it compromises the run-time stack of a process.



```
#include <string.h>
void foo (char *bar)
{
  char  c[12];
  strcpy(c, bar);
}

int main (int argc, char **argv)
{
  foo(argv[1]);
  return(1);
}
```

[10]

(iii) How can the Hamming (7, 4) code be used to encode 8-bit data? What error correcting and error detecting capabilities does the resulting encoding scheme have?

[12]

(iv) Consider you are a member of a development team responsible for the control software of a washing machine. The software is controlled by a scheduler based on the cyclic executive approach. It controls 3 tasks with cycle times of 50 msec, 75 msec and 100 msec as shown in Table 1 below. The execution time of each task is well below its cycle time.

Task	Period (msec)	Exec Time (msec)
A	50	10
B	75	10
C	100	10

Table 1

Section continues on next page.

Using (pseudo and/or C) code, prototype a timer-interrupt-controlled cyclic executive for the above task schedule. Your answer must include an implementation of the timer-interrupt service routine.
Also, task A overruns occasionally (e.g. its execution time is about 60 msec). Enhance your cyclic executive to detect such violations.

[12]

Q5

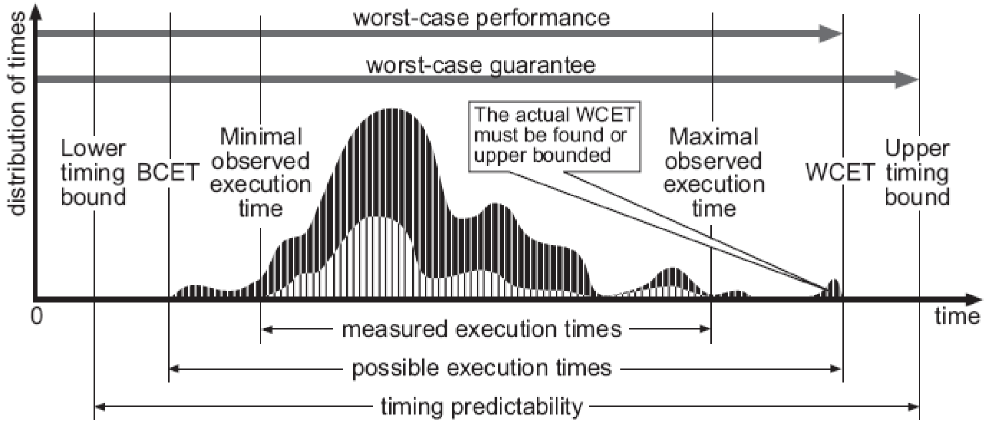
(i) Briefly discuss the concept of N-modular hardware redundancy. Furthermore, provide an example for a redundant system that consists of
- N=3 independent components that have a failure probability of 10% each and
- a 2v3 voter.
Calculate the overall system reliability.

[8]

(ii) Using examples show how (a) a 2-disks RAID-0, (b) a 2-disks RAID-1, and (c) a 4-disks RAID-5 system respond to the failure of an entire hard disk.

[10]

(iii) The diagram below illustrates the Worst Case Execution Time (WCET) for a sample task under both offline code analysis (dark shaded) and actual runtime testing (light shaded). Explain each of the indicated points and ranges in the diagram and outline the importance of such analysis in RTS design.



[10]

Section continues on next page.

(iv) Using the task set in Table 2 below:
- Calculate the overall *CPU utilisation U*.
- Determine the process schedule using the *RM scheduling algorithm*.
- Determine the process schedule using the *EDF scheduling algorithm*.
- Comment on the difference between both schedules.

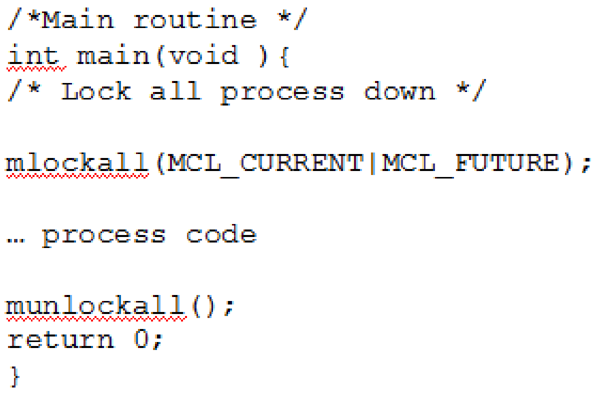
Task	Execution Time	Period
1	10	50
2	20	100
3	20	150
4	80	200

Table 2

[12]

Q6

(i) Memory locking is an important feature of POSIX 4. Use the code snippet below and the 7-state process model to explain why this feature is important for (hard) real-time systems.



```
/*Main routine */
int main(void ){
/* Lock all process down */

mlockall(MCL_CURRENT|MCL_FUTURE);

… process code

munlockall();
return 0;
}
```

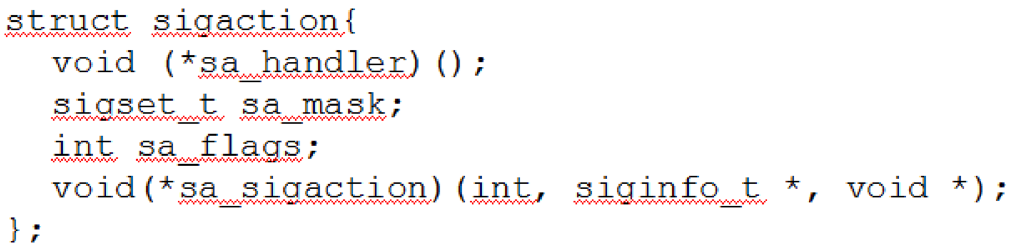
[10]

(ii) What is meant by the *priority inversion problem* and how can it be solved? Use an example to illustrate your answer.

[12]

Section continues on next page.

(iii) Summarise the main features of POSIX signals. Furthermore, outline the purpose and functionality of the structure *sigaction* below and show how this structure allows the handling of queued signals as defined in POSIX.4.



```
struct sigaction{
   void (*sa_handler)();
   sigset_t sa_mask;
   int sa_flags;
   void(*sa_sigaction)(int, siginfo_t *, void *);
};
```

[10]

(iv) Discuss how static software redundancy can be provided via N-version programming. What are the advantages and what are potential pitfalls of this approach? Furthermore, briefly elaborate on different voter types that can be used.

[8]